CS 134: Networks Problem set 8

Prof. Yaron Singer Due: **Wednesday 4/5/2017**

1. Alternative Definition for Submodular Functions (15 points) Recall that in class we said that a set function $f : 2^V \to \mathbb{R}$ is *submodular* if for every $S, T \subseteq V$ for which $S \subseteq T$ and $a \notin T$ we have that:

$$f(S \cup a) - f(S) \geq f(T \cup a) - f(T)$$

a. **(7 points)** Show that f is a submodular function **if and only if** (iff) for every $A, B \subseteq V$ we have that:

$$f(A \cup B) \leq f(A) + f(B) - f(A \cap B)$$

As a reminder, A iff B if both A implies B and B implies A.

b. **(8 points)** Show that every submodular function is subadditive, i.e. if if f is submodular then $\forall A, B \subseteq V$:

$$f(A \cup B) \leq f(A) + f(B)$$

2. The Transition Matrix (20 points) Given a graph $G = (V, E)$, for any $u \in V$, let $\mathcal{N}(u)$ denote the set of neighbors of u, including u itself, and $d(u) = |\mathcal{N}(u)|$. Recall that the *transition matrix* of G is the $n \times n$ matrix M defined as:

$$M_{u,v} = \begin{cases} 1/d(u) & v \in \mathcal{N}(u); \\ 0 & \text{otherwise} \end{cases}$$

Consider the following network

$$G = (\{a, b, c, d, e, \}, \{ab, bc, cd, de, ac, ad, ae, aa, bb, cc, dd, ee\}).$$

a. **(6 points)** Construct the transition matrix for G.

b. **(7 points)** What is the probability of a random walk going from node a to node e in two steps?

c. **(7 points)** In the voter model, what is the probability that the opinion of node a reaches e in two steps?

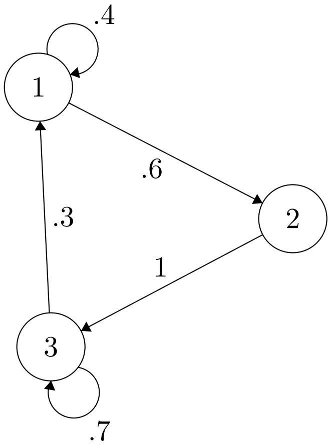
3. Random Walks (10 points) Prove that for a graph $G = (V, E)$ with transition matrix M, for any $u, v \in V$ the probability of a random walk starting at u and ending at v after t steps is $\mathbf{1}_u M^t \mathbf{1}_v^\top$ where $\mathbf{1}_x$ denotes the row vector that takes the value 1 in the index that corresponds to the node $x \in V$ and 0 in all other indices.

4. DeGroot Model (30 points) Consider a group of n friends. At time $t = 0, 1, 2, \ldots$ each person i has an opinion, $p_i(t) \in [0, 1]$, of the new James Bond movie. Every period they talk to each other and exchange opinions. The *DeGroot model* posits that person i puts weight $m_{ij} \geq 0$ on person j's opinion (putting weight on one's own opinion is allowed), subject to $\sum_{j=1}^{n} m_{ij} = 1$. In other words, if in period t, the people's opinions are $(p_1(t), ..., p_n(t))$, then i's opinion in period $t+1$ is

$$p_i(t+1) = \sum_{j=1}^{n} m_{ij} p_j(t) \tag{1}$$

a. **(3 points)** Show that Eq. (1) can be rewritten as $p(t+1) = Mp(t)$, where $p(t+1)$ and $p(t)$ are vectors of length n and M is some $n \times n$ matrix.

b. **(3 points)** Consider the network given in the picture below. An arrow from node i to node j means that person i puts weight on j's opinion. Write down M.



c. **(4 points)** It turned out that person 1 loved the movie, person 2 thought it was pretty good, and person 3 didn't get it: $p(0) = (1, 0.7, 0)$. Find $p(1)$ and $p(2)$.

d. **(6 points)** What are the limiting opinions? In other words, find $\lim_{t\to\infty} p(t)$. Does the group reach a consensus? Give an intuitive explanation of why consensus is reached.

e. **(7 points)** Give an example of a network with $n = 3$ with no isolated nodes, and a vector of initial opinions, such that a consensus is never reached – that is, so the sequence of vectors $(p(t))_{t=1}^{\infty}$ has no limit.

f. **(7 points)** Suppose now agents are arranged in a ring, numbered $1, 2, \ldots, n$ with person i putting 1/2 of her updating weight on himself ($m_{ii} = 1/2$) and 1/2 of her weight on his clockwise neighbor($m_{i,i+1} = 1/2$). Indices are read modulo n.

What will opinions converge to now? I.e., write down an explicit formula for $p(\infty)$, which is defined to be $\lim_{t\to\infty} p(t)$. This formula should involve only references to entries of $p(0)$ and n. This can be done with little calculation and largely through reasoning.

5. Learning Influence Locally (again) (25 points) (If you don't like backstories in your psets, feel free to skip down a couple paragraphs. This problem is similar to problem 5 from problem set 6, but with a new diffusion model—something very similar to the Voter model.)

Raynor's been hired by a hot new social network called BetaPatch. They're not entirely sure what they do, but they know they only want the most popular, influential people using their app, so they'd like to figure that out who that is, and they want Raynor to take care of it.

Thanks to some lengthy field work on Facebook and a very divisive poll about local Mexican eateries in Harvard Square, Raynor has some data of the student network and how opinions change in the student community over time. He now wants to work backwards from this data using the Voter model as an assumption to derive popularity of students.

As in problem set 6, let's say we have a graph $G = (V, E)$ and some opinion data for each node $u \in V$. Assume each node (student) has an opinion that changes over time based on the influence of its neighbors (Instagram accounts) for a particular observed diffusion process (procrastination in the library). For a particular observed diffusion process i, let $opinion_i$ be a vector of timestamped opinions, where $opinion_i[u]$ is a vector such that $opinion_i[u][t]$ is equal to the opinion of node u at time t during observed cascade i, where t ranges from 0 to $\tau - 1$ so that $opinion_i[u]$ has length τ, where opinions are either 0 (node is inactive) or 1 (active).

Our model is the same as the Voter model, but using a deterministic threshold rather than a probabilistic one. In this model, each directed edge (u, v), has some corresponding edge weight $w_{(u,v)} \in (0, 1]$. At each time step t, node u will look at each of its neighbors and update its opinion to reflect the majority "weighted" opinion—that is, if $\sum\limits_{v \in N(n)} w_{(u,v)} o_{t-1}(v) \geq .5$, then u will update its opinion to be 1 at time step t, and otherwise, will update its opinion to be 0 at time step t, where $o_{t-1}(v)$ is the opinion of node v at time step $t-1$ and is either 0 or 1. How can we help Raynor out by describing this situation using a linear program such that solving said linear program will give us an approximation for the edge weights?

If we want to fit the Voter model locally to the opinion data, we can learn each of the edge weights $w_{(u,v)}$ in the graph by modeling the problem as binary classification and solving using a linear program. Consider a node n. We're trying to learn each of the edge weights $w_{(u,v)}$ for $v \in N(u)$. If we have some opinion data, and we know that at some time step t, u has opinion 0, and at time step $t + 1$, u has opinion 1, we know that the sum of the edge weights to u's neighbors that had opinion 0 at $t - 1$ was $\geq .5$ at $t - 1$, but was $< .5$ at t.

The file *pset8_network.txt* is a directed graph, where a b corresponds to an edge from a to b. The file *node_opinions.pk* is an array of opinions, where each entry *node_opinions*$[i]$ is a dictionary equivalent to the vectors $opinion_i$ described above, such that $opinion_i[n]$ gives the vector of timestamped opinions described.

a. **(1 point)** Load the file *pset8_opinions.pk*. If you're using python, you can use python's 'pickle' library and the 'load' function to easily turn the file into an array; if you're using another language, we've provided *node_opinions.json*, a JSON-encoded file that you can easily load in any language using the standard JSON encoding and the respective JSON library for your language (though you might have to do some additional string to integer conversion). What is τ and how many different diffusion processes are there?

b. **(1 point)** Now load the file *pset8_network.txt*. How many nodes are there? What is the average (out-)degree?

c. **(10 points)** Because he tends to have to debug a lot, Raynor first wants to make sure he has the right setup by writing down how influence will affect opinions. Using the definition of the Voter model described in lecture, and given a node u, write the linear program that, given d diffusion processes with τ time steps each, uses all available information to approximate the edge weights $w_{(u,v)}$ for $v \in N(u)$. That is, write below the set of linear equations that constrain how a node in the voter model influences its neighbors.

Feel free to use indices to refer to time steps and cascades, but **be clear with all your notation and what means what.** (Raynor won't be able to read messy variables! It might even be helpful to describe what each individual equation refers to.) Don't forget to also include constraints such as the fact that all the edge weights have to be nonnegative, or that the weights for a single node should add up to one.

d. **(10 points)** Time for Raynor to get coding–or rather, for you to get coding. Using the linear program you described above, solve the program for each node and approximate the edge weights of each $e \in E$. Raynor is lazy, so to solve the program, you can use any one of the many linear program solver packages available (for python, scipy has a linear programming package).

Note that linear program solver packages will find variables that satisfy the constraints provided and fit some maximization or minimization constraint. **For the sake of making everything standardized for the grader, just maximize the sum of all the variables you're calculating in the linear program**. Additionally, since we know that the Voter model can be fit to the data (since the data was generated by the Voter model! Raynor's data is precise), we won't need to worry about slack variables.

What's the estimated weight of edge $(2, 16)$? Of $(29, 22)$? Additionally, submit your estimates as a separate file formatted as a .csv, where entries in the first column correspond to the first node in the edge, entries in the second column correspond to the second node in the edge, and entries in the third column correspond to the edge weight.

e. **(3 points)** Assuming popularity is the average of influence, what five students (nodes) should Raynor first invite to the app and what's their average influence? You should use edge weights as measures of influence, but consider the Voter model and be careful about the direction of your edges!...